

Lab 1_1 보고서

20210479 이주현

① 개요

Lab 1_1는 Xilinx Vivado 개발 환경에 적응하고, Verilog 문법의 기초를 다지기 위한 활동이다. 이를 위해 AND gate를 직접 구현해보고, 이것의 작동을 확인하기 위한 testbench를 작성하고, 실제 시뮬레이션을 구동하는 것을 목표로 한다.

② 이론적 배경

(1) HDL

HDL이란 Hardware Description Language의 줄임말로, 디지털 시스템을 표현, 설계, 구현, 합성, 테스트, 문서화하기 위한 언어이다. 이 Lab 과제에서는 HDL을 기술하기 위한 프로그래밍 언어로 Verilog를 다루며, 동작/구조의 표현 방식 중 Gate level modeling을 학습한다.

(2) Gate Level Modeling

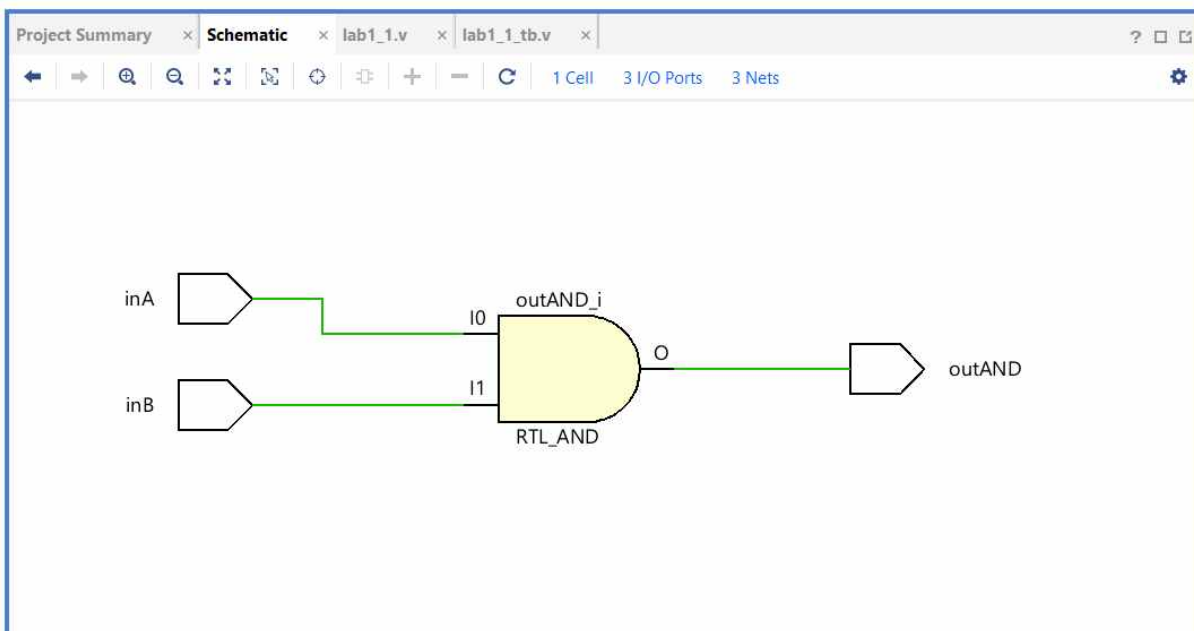
Gate level modeling이란 기초 연산인 and, or, not, nand 따위가 함수식의 형태로 구현되어 있는 것을 의미한다. 회로의 구성과 비슷하여 보다 직관적으로 이해가 가능하다는 장점이 있다.

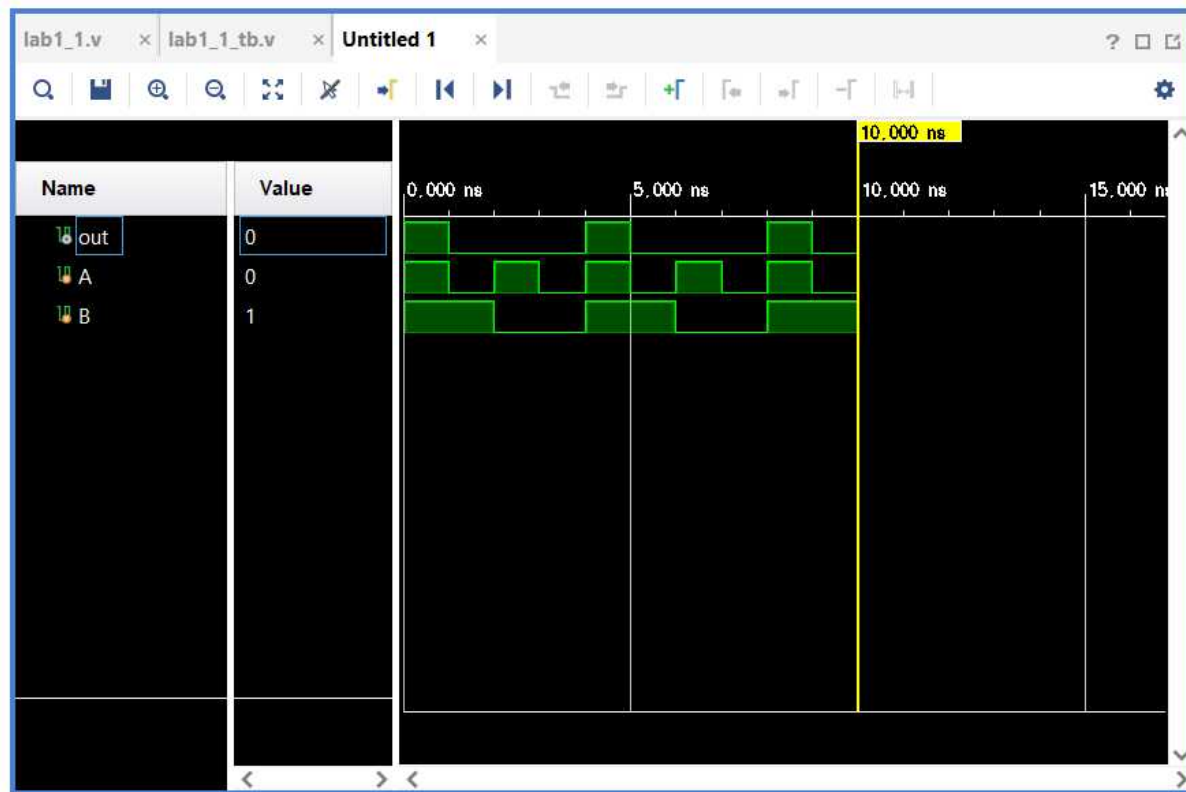
③ 실험 준비

(1) 오리엔테이션 및 Lab0에서 구현한 OR gate를 통해 기본적인 Vivado 사용법과 문법, Testbench 작성 방법을 숙지한다.

④ 결과

작성한 회로도(RTL analyzer schematic)와 시뮬레이션 결과는 다음과 같다. 이로 미루어 보았을 때 구현이 올바르게 된 것을 알 수 있다.





5 논의

이번 과제를 통해 Verilog의 개발 환경에 적응하고, 문법을 익힐 수 있었다. 기본적인 Gate level modeling의 방법에 대해서도 알 수 있었다.

Lab 1_2_i 보고서

20210479 이주현

1 개요

Lab 1_2_i는 Vivado 개발 환경에 적응하고, Verilog 문법을 익히기 위한 활동이다. 나아가 Functionally complete set의 개념을 숙지하고 이를 구현해보는 것을 목표로 한다.

2 이론적 배경

(1) HDL

HDL이란 Hardware Description Language의 줄임말로, 디지털 시스템을 표현, 설계, 구현, 합성, 테스트, 문서화하기 위한 언어이다. 이 Lab 과제에서는 HDL을 기술하기 위한 프로그래밍 언어로 Verilog를 다루며, 동작/구조의 표현 방식 중 Gate level modeling을 학습한다.

(2) Functionally Complete Set

모든 Bool algebra를 표현할 수 있는 어떤 논리 연산들의 집합을 이르는 말이다. 기초 연산으로 이루어진 {and, or, not} 등이 Functionally complete set에 해당한다.

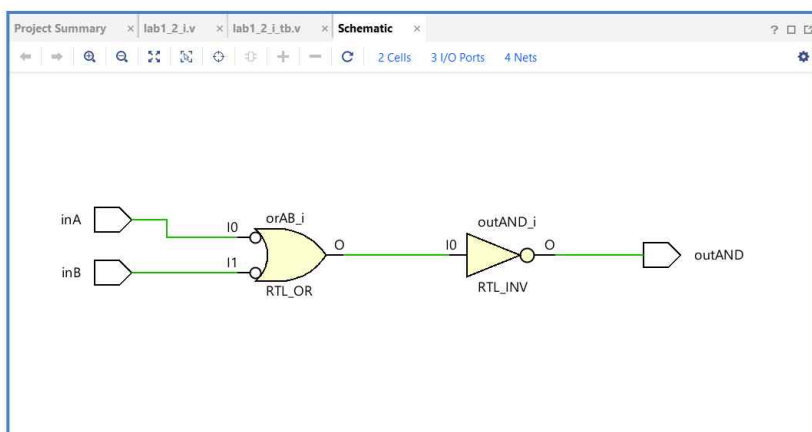
3 실험 준비

(1) 기본적인 Vivado 사용법과 문법, Testbench 작성 방법을 숙지한다.

(2) or과 not을 이용해 $AB = (A' + B)'$ 과 같은 모습으로 and 연산을 표현할 수 있다. 이 회로를 구성할 수 있도록 코드를 작성한다.

4 결과

작성한 회로도(RTL analyzer schematic)는 다음과 같다. 이로 미루어 보았을 때 구현이 올바르게 된 것을 알 수 있다.



5 논의

이 과제를 통해 Functionally complete set을 만들기 위해서는 기초 연산으로 이루어진 집합인 {and, or, not}을 만족하면 된다는 것을 알 수 있었다. 또한 그 집합을 구현하기 위해 {or, not}을 통해 and를 구현하는 방법을 알 수 있었다.

Lab 1_2_ii 보고서

20210479 이주현

1 개요

Lab 1_2_ii는 Functionally complete set의 개념을 숙지하고, 이 개념의 응용 예시로써 {and, not} 집합으로 or 연산을 구현하여 Functionally complete set을 만들어 보는 것을 목표로 한다.

2 이론적 배경

(1) HDL

HDL이란 Hardware Description Language의 줄임말로, 디지털 시스템을 표현, 설계, 구현, 합성, 테스트, 문서화하기 위한 언어이다. 이 Lab 과제에서는 HDL을 기술하기 위한 프로그래밍 언어로 Verilog를 다루며, 동작/구조의 표현 방식 중 Gate level modeling을 학습한다.

(2) Functionally Complete Set

모든 Bool algebra를 표현할 수 있는 어떤 논리 연산들의 집합을 이르는 말이다. 기초 연산으로 이루어진 {and, or, not} 등이 Functionally complete set에 해당한다.

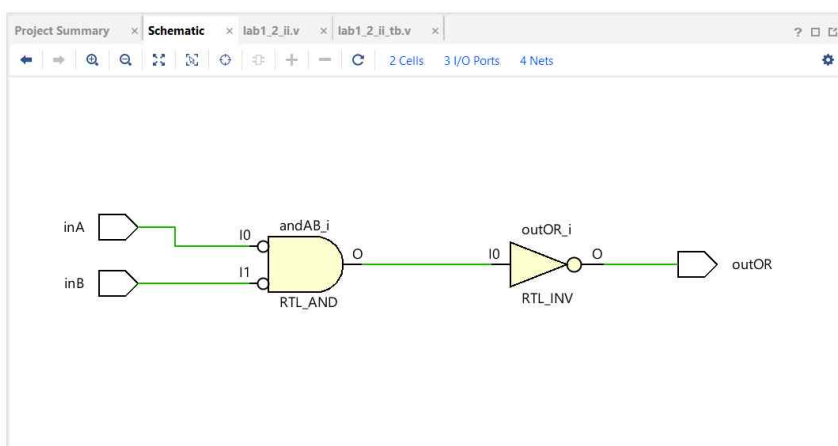
3 실험 준비

(1) 기본적인 Vivado 사용법과 문법, Testbench 작성 방법을 숙지한다.

(2) and와 not을 이용해 $A+B = (A'B)'$ 과 같은 모습으로 or 연산을 구현할 수 있다. 이 회로를 구성할 수 있도록 코드를 작성한다.

4 결과

작성한 회로도(RTL analyzer schematic)는 다음과 같다. 이로 미루어 보았을 때 구현이 올바르게 된 것을 알 수 있다.



5 논의

이 과제를 통해 Functionally complete set을 만들기 위해서는 기초 연산으로 이루어진 집합인 {and, or, not}을 만족하면 된다는 것을 알 수 있었다. 또한 그 집합을 구현하기 위해 {and, not}을 통해 or을 구현하는 방법을 알 수 있었다.

Lab 1_2_iii 보고서

20210479 이주현

1 개요

Lab 1_2_iii는 Functionally complete set의 의미를 익히고, 응용 예시의 하나로 {nand} 집합만을 이용하여 {and, or, not} 집합을 구현해보는 것을 목표로 한다.

2 이론적 배경

(1) HDL

HDL이란 Hardware Description Language의 줄임말로, 디지털 시스템을 표현, 설계, 구현, 합성, 테스트, 문서화하기 위한 언어이다. 이 Lab 과제에서는 HDL을 기술하기 위한 프로그래밍 언어로 Verilog를 다루며, 동작/구조의 표현 방식 중 Gate level modeling을 학습한다.

(2) Functionally Complete Set

모든 Bool algebra를 표현할 수 있는 어떤 논리 연산들의 집합을 이르는 말이다. 기초 연산으로 이루어진 {and, or, not} 등이 Functionally complete set에 해당한다.

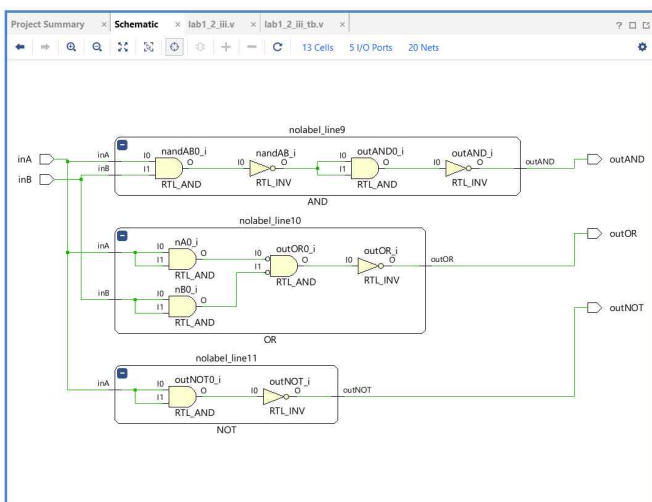
3 실험 준비

(1) 오리엔테이션 및 Lab0에서 구현한 OR gate를 통해 기본적인 Vivado 사용법과 문법, Testbench 작성 방법을 숙지한다.

(2) nand로 세 연산을 각각 $and \rightarrow ((AB))'$, $or \rightarrow (A'B)'$, $not \rightarrow (AA)' = A'$ 와 같이 나타낼 수 있다. 이 회로를 구성할 수 있도록 코드를 작성한다.

4 결과

작성한 회로도(RTL analyzer schematic)는 다음과 같다. 이로 미루어 보았을 때 구현이 올바르게 된 것을 알 수 있다.



5 논의

이 과제를 통해 Functionally complete set을 알 수 있었고, nand를 통해 세 연산을 구현하는 방법을 알게 되었다.

Lab 1_2_iv 보고서

20210479 이주현

1 개요

Lab 1_2_iv는 Functionally complete set의 개념을 숙지하고, {nor} 집합을 이용해 Functionally complete set의 하나인 {and, or, not} 집합을 직접 구현해보는 것을 목표로 한다.

2 이론적 배경

(1) HDL

HDL이란 Hardware Description Language의 줄임말로, 디지털 시스템을 표현, 설계, 구현, 합성, 테스트, 문서화하기 위한 언어이다. 이 Lab 과제에서는 HDL을 기술하기 위한 프로그래밍 언어로 Verilog를 다루며, 동작/구조의 표현 방식 중 Gate level modeling을 학습한다.

(2) Functionally Complete Set

모든 Bool algebra를 표현할 수 있는 어떤 논리 연산들의 집합을 이르는 말이다. 기초 연산으로 이루어진 {and, or, not} 등이 Functionally complete set에 해당한다.

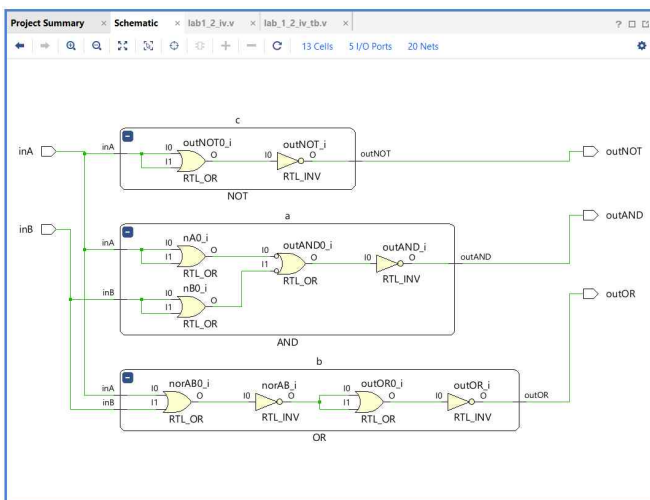
3 실험 준비

(1) 오리엔테이션 및 Lab0에서 구현한 OR gate를 통해 기본적인 Vivado 사용법과 문법, Testbench 작성 방법을 숙지한다.

(2) nor로 세 연산을 각각 $\text{and} \rightarrow (A' + B)'$, $\text{or} \rightarrow ((A+B))'$, $\text{not} \rightarrow (A+A)'$ 와 같이 나타낼 수 있다. 이 회로를 구성할 수 있도록 코드를 작성한다.

4 결과

작성한 회로도(RTL analyzer schematic)는 다음과 같다. 이로 미루어 보았을 때 구현이 올바르게 된 것을 알 수 있다.



5 논의

이 과제를 통해 Functionally complete set을 알 수 있었고, nor를 통해 세 연산을 구현하는 방법을 알게 되었다.